

Dynamical Adaptation in ORNs

Srinivas Gorur-Shandilya

July 1, 2014

Effect of Correlation Time on Linear Filters

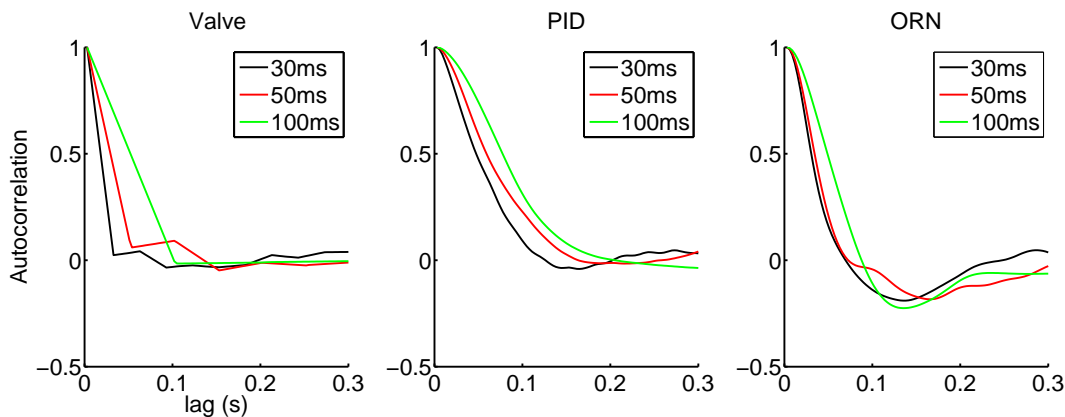
What is the effect of correlation time of the stimulus on the ORN response, and the filter that we calculate from this data? We have data where the same class of ORN responds to flickering stimuli with different correlation lengths: 30ms, 50ms and 100ms. The filters extracted from these datasets are slightly different, and fitted output non-linearities seem to have slopes that decrease with increasing correlation time.

Contents

- Are filters different for different correlations?
- Cross-prediction using different filters
- Synthetic data: filter and non-linearity estimation

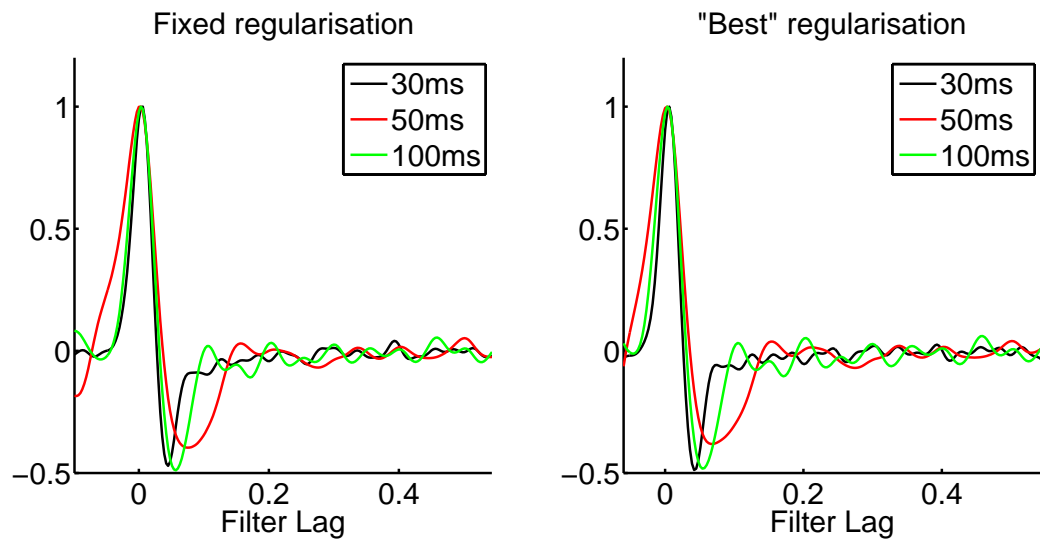
Is this a real phenomenon? Why does this happen?

First, we plot the autocorrelation functions of the valve, the stimulus and the ORN and check that they make sense.

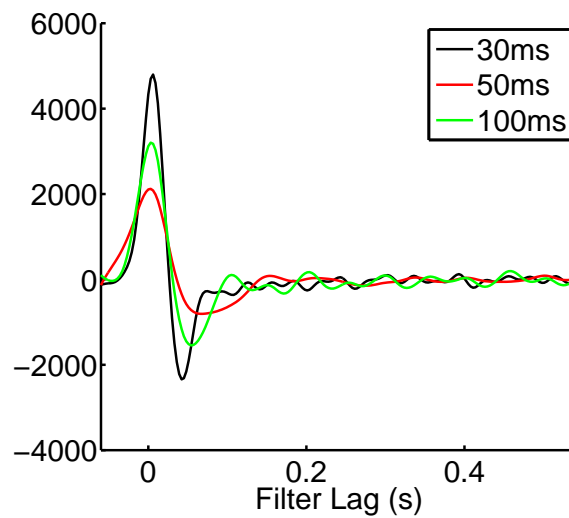


Are filters different for different correlations?

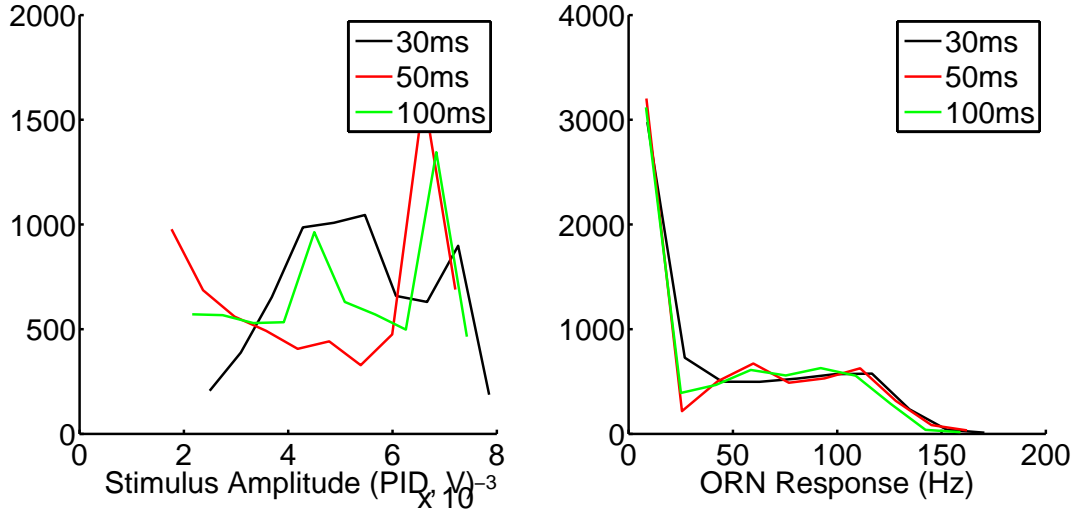
Now we fit filters to these data sets. In the figure below, the filters on the left are calculated using a fixed regularisation (of 1), and the filters on the right, the regularisation is varied in each case to find the "best" filter, minimising errors, high frequency components, and having a gain as close to unity as possible. In either case, we see the filters are slightly different for the three data sets; the correlation time of the stimulus seems to affect the filter we extract.



These filters differ not only in their shape but also their amplitude, which is a simple scaling because the input statistics are different.

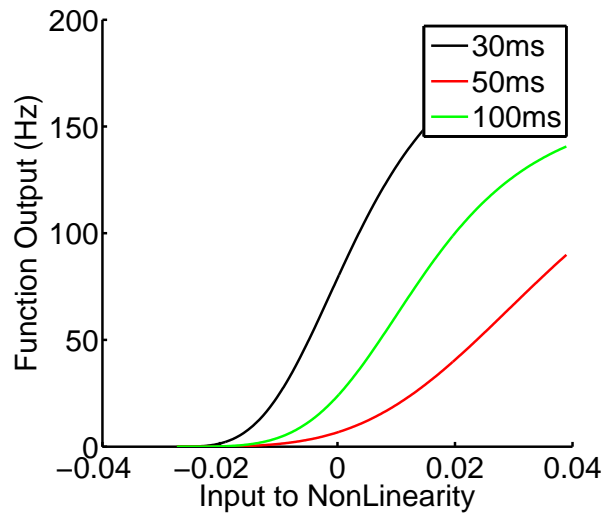


Why does the filter amplitude change so much? let's look at the stimulus and the response histograms:



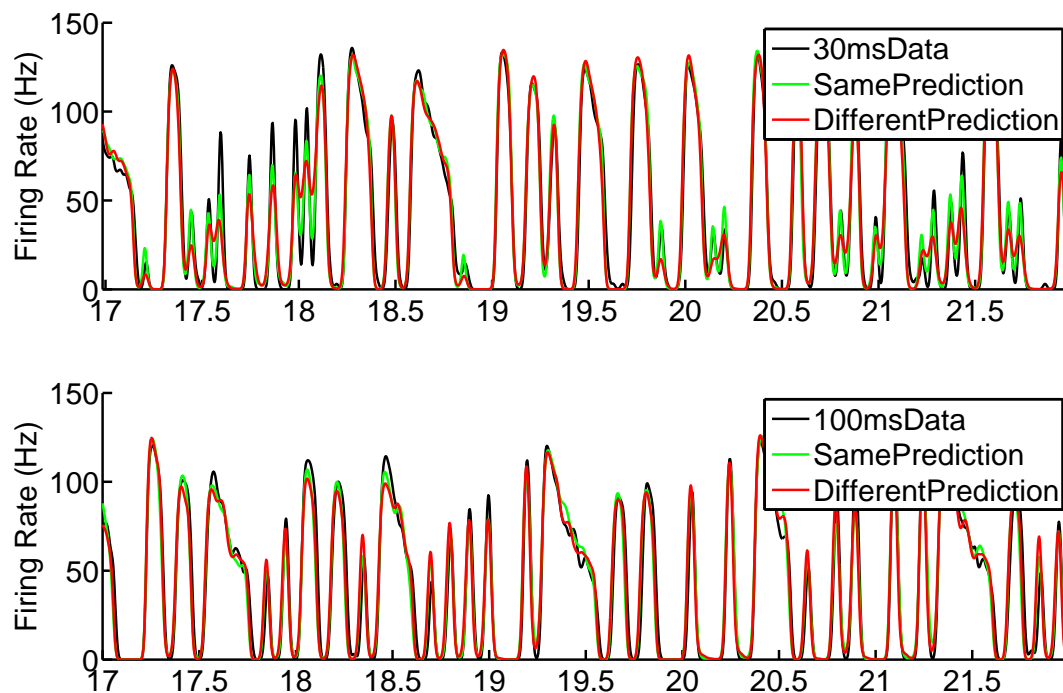
Cross-prediction using different filters

To disentangle the effects of filter amplitude from possible changes in filter shape, we fit a static non-linearity to the output of each linear filter and use that to predict the output.



No monotonic relationship is observed between correlation length and slope of fitted non-linearity.

How good are these filters at estimating the data? In the figure we below, we show the two extreme cases (the 30ms and 100ms correlated data) and the predictions from the filters calculated from their own data sets, along with predictions from filters calculated using the others' data. The cross-prediction filter outputs are then passed through the corresponding static non-linearities,



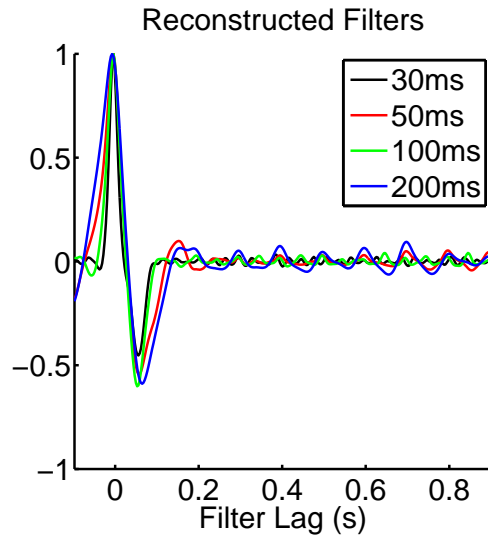
How good are these cross-predictions? For the three data sets we have, we measure how well the filter from each data set predicts the data. In the matrix below, the element in i th row and j th column shows the r -square of linear prediction by the i th filter to j th data set, where the first data set has 30ms correlated stimulus, the second has 50ms correlated stimulus, and the 3rd has 100ms correlated stimulus.

0.9344	0.9638	0.9424
0.8271	0.9486	0.8788
0.8991	0.9601	0.9397

In particular, the 30ms filter outperforms all other filters, even the "native" filters of the 50ms and 100ms data.

Synthetic data: filter and non-linearity estimation

What effect does using stimuli with various correlation lengths have on our estimation of the filter? Here we use a "fake" filter to represent the neuron, and pass stimuli with various correlation lengths to it, and then back out the filter from the synthetic data output.



We see that the estimated filter gets wider for longer correlation lengths, even though we know that the actual filter is constant. Thus, trying to estimate filters of neurons with longer correlation times can back out longer-than-actual timescales in the filter. This is because the response of the system to fast time scales isn't being probed at all.